SH7216 Group

Reprogramming Flash Memory in User Program Mode Using an Ethernet Connection

R01AN0289EJ0211
Rev.2.11
Apr 06, 2011

Introduction

This application note presents a sample program that uses the Ethernet controller (EtherC) and Ethernet controller direct memory access controller (E-DMAC) of the SH7216 to receive data from a host PC via an Ethernet connection and then writes the data to the on-chip flash memory of the SH7216.

The simple flash API (standard API) for the SH-2 and SH-2A, supplied by Renesas Electronics, is used to reprogram the on-chip flash.

Target Device

SH7216

Contents

1. Introduction 2
2. Specifications of Sample Programs 3
3. Using the Sample Program 26
4. Notes on Use of Flash Memory (ROM) 26
5. Reference Documents 27

1. Introduction

1.1 Specifications

In the sample application the user program mode of the SH7216 is used to write data to the user MAT in flash memory (ROM). The data used to program the ROM is transferred from the host PC via an Ethernet link.

The application that runs on the host PC uses a socket interface to transfer data by means of the TCP/IP protocol.

The sample program that runs on the SH7216 uses the Ethernet controller (EtherC) and Ethernet controller direct memory access controller (E-DMAC) to receive the data transferred with the TCP/IP protocol. Then it writes the data to the user MAT in the ROM.

Figure 1 shows the system configuration of the sample application.

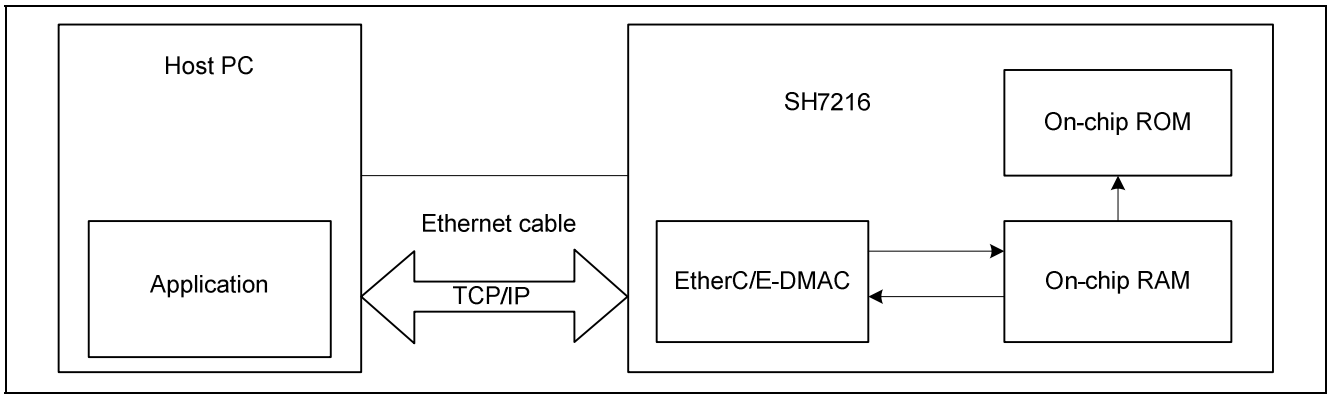


Figure 1 System Configuration

1.2 Functions Used

- Ethernet controller (EtherC)
- Ethernet controller direct memory access controller (E-DMAC)
- Flash memory (ROM)
- On-chip RAM
- Compare match timer (CMT)

1.3 Applicable Conditions

MCU	SH7216
Operating frequency	Internal clock: 200 MHz Bus clock: 50 MHz Peripheral clock: 50 MHz
Integrated development environment	Renesas Electronics High-performance Embedded Workshop, Ver. 4.07.00.007
C compiler	Renesas Electronics SuperH RISC engine Family C/C++ Compiler Package, Ver. 9.03, Release 02
Compile options	High-performance Embedded Workshop default settings (-cpu=sh2afpu-pic=1-object="$(CONFIGDIR)\$(FILELEAF).obj" -debug-gbr=auto-chgincpsth-errorpath-global_volatile=0 -opt_range=all-infinite_loop=0-del_vacant_loop=0 -struct_alloc=1-nologo)

1.4 Related Application Notes

- H8S/2472 and SH7216: uIP TCP/IP Protocol Stack Demonstration
- SH7216 Group: Using RCAN-ET in User Program Mode
- SH7216 Group: Using IIC3 in User Program Mode
- SH7216 Group: Using RSPI in User Program Mode
- SH Family: Simple Flash API for SH-2 and SH-2A

2. Specifications of Sample Programs

The specifications of the sample program for the SH7216 and the sample program for the host PC used in this sample application are described in this section.

2.1 Overall Sequence

Figures 2 and 3 show the operation sequence of the sample program for the host PC and the sample program for the SH7216. The arrows between the host PC and the SH7216 indicate TCP data transfers.

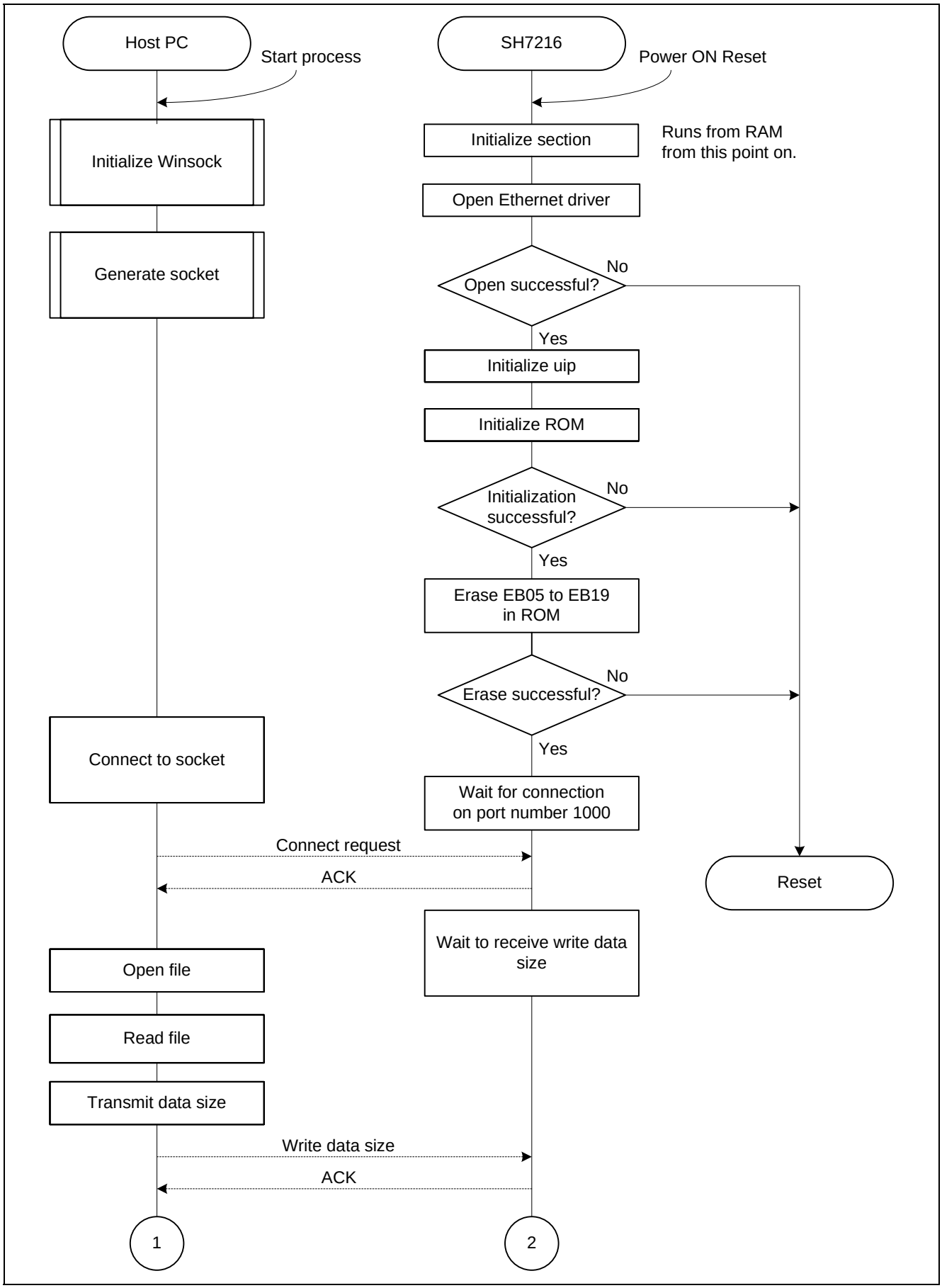


Figure 2 Overall Sequence (1/2)

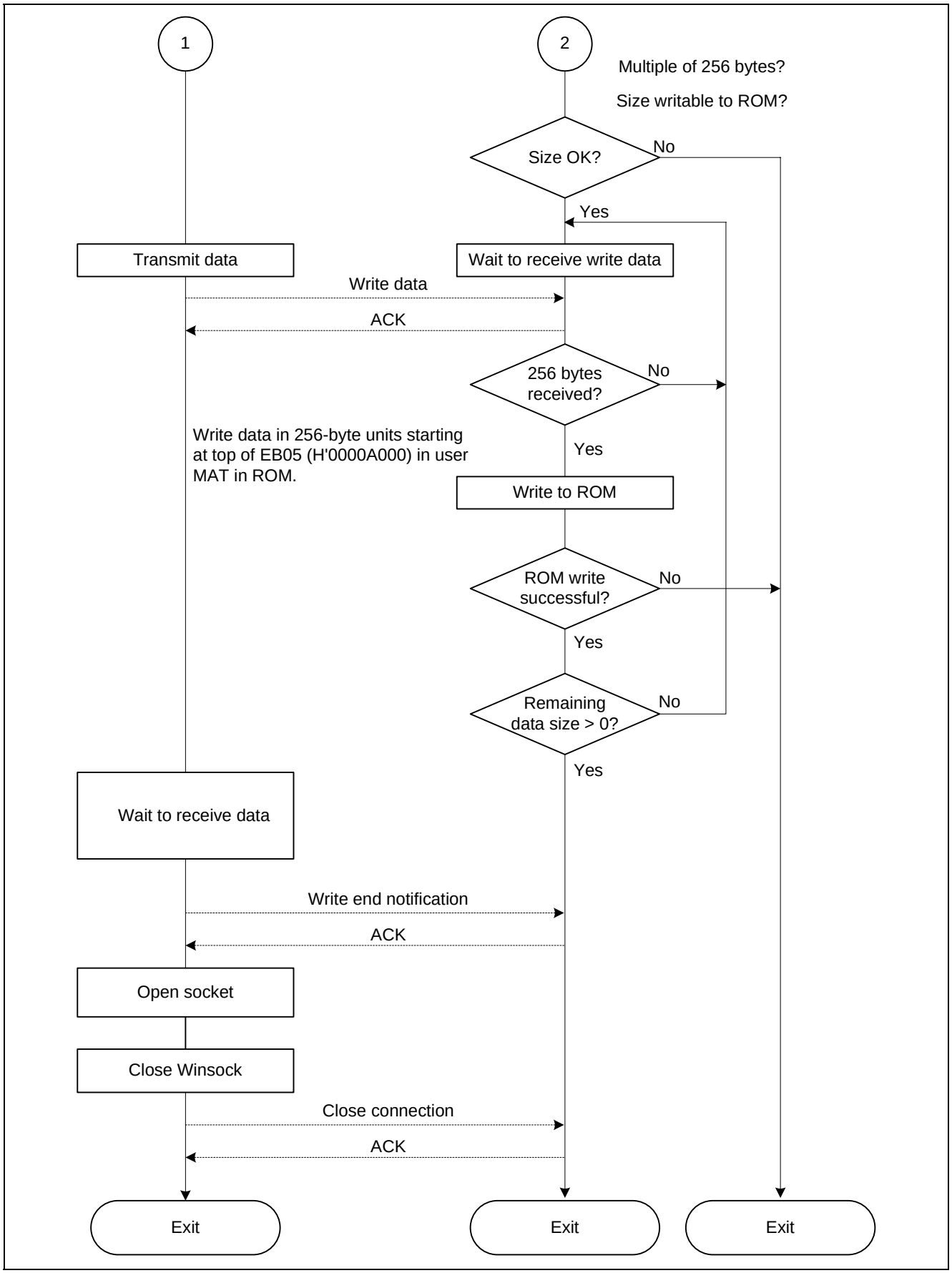
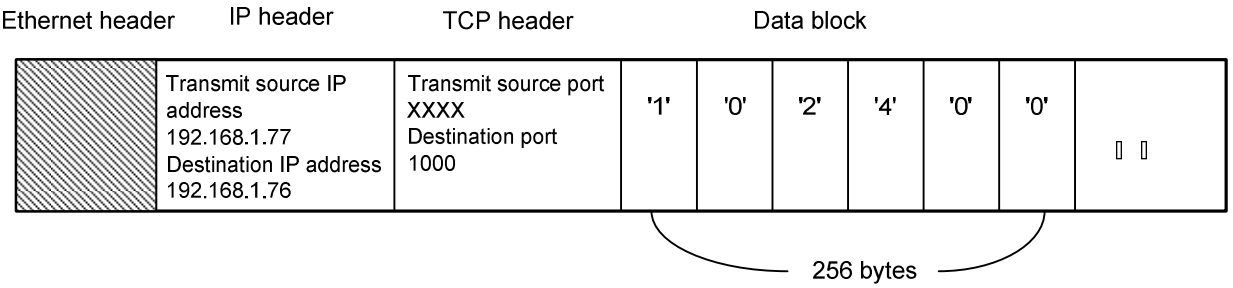


Figure 3 Overall Sequence (2/2)

Figure 4 shows the command format for the write data size and write end notification.

• Write data size

The size of the data block is 256 bytes. If the write data size is 1,024 bytes, “1024” is stored in the data block as ASCII character codes. The remaining space is filled with zeros.



• Write end notification

When processing completes normally, “The flash writing was completed.” is stored in the data block as ASCII character codes. The size of the data block is 32 bytes.

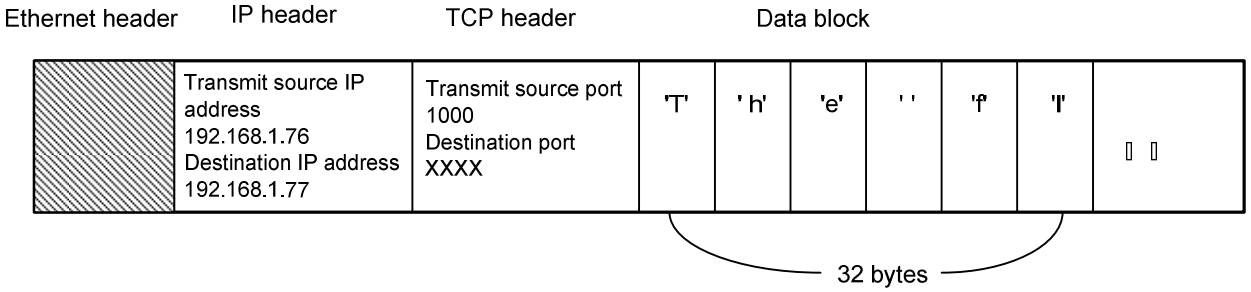


Figure 4 Command Format

2.2 Sample Program for SH7216

The sample program for the SH7216 comprises the main process, flash write controller, TCP/IP protocol stack, Ethernet driver, and flash memory driver.

The program area, constant area, and initialization data area of the program are allocated within blocks EB00 to EB04 (addresses H'00000000 to H'00009FFF) of the user MAT in ROM.

The SH7216 cannot use a program allocated to the user MAT to program or erase the user MAT. Therefore, this program transfers the contents of the program area, constant area, and initialization data area to the on-chip RAM (addresses H'FFF80000 to H'FFF8FFFF) immediately after the power-on reset, and the program runs from the on-chip RAM from that point onward.

The sample program for the SH7216 functions as a server. The IP address and port number are fixed at 192.168.1.76 and 1000, respectively.

Figure 5 shows the structure of the sample program for the SH7216. The arrows indicate the direction of control.

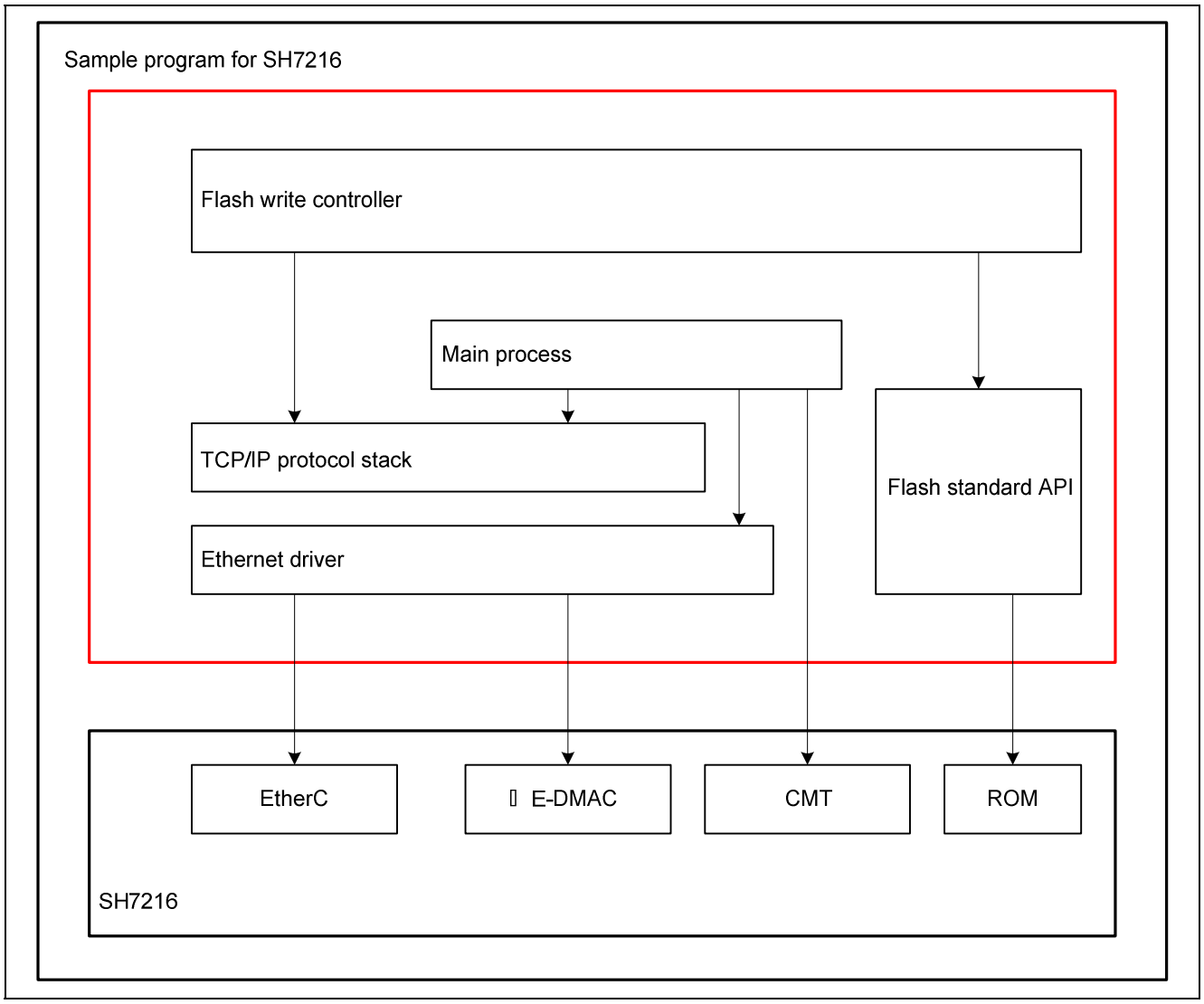


Figure 5 Structure of Sample Program for SH7216

2.2.1 Main Process

In order to port uip to the SH7216, it is necessary for the main loop to perform the following two tasks.

- Checking whether or not an Ethernet frame has arrived from the network
- Checking whether the uip timer has timed out

If an Ethernet frame has arrived, uip_input() must be called to notify uip. In like manner, if the uip timer has timed out, uip_periodic() must be called to notify uip.

Figures 6 to 9 are flowcharts of the main process.

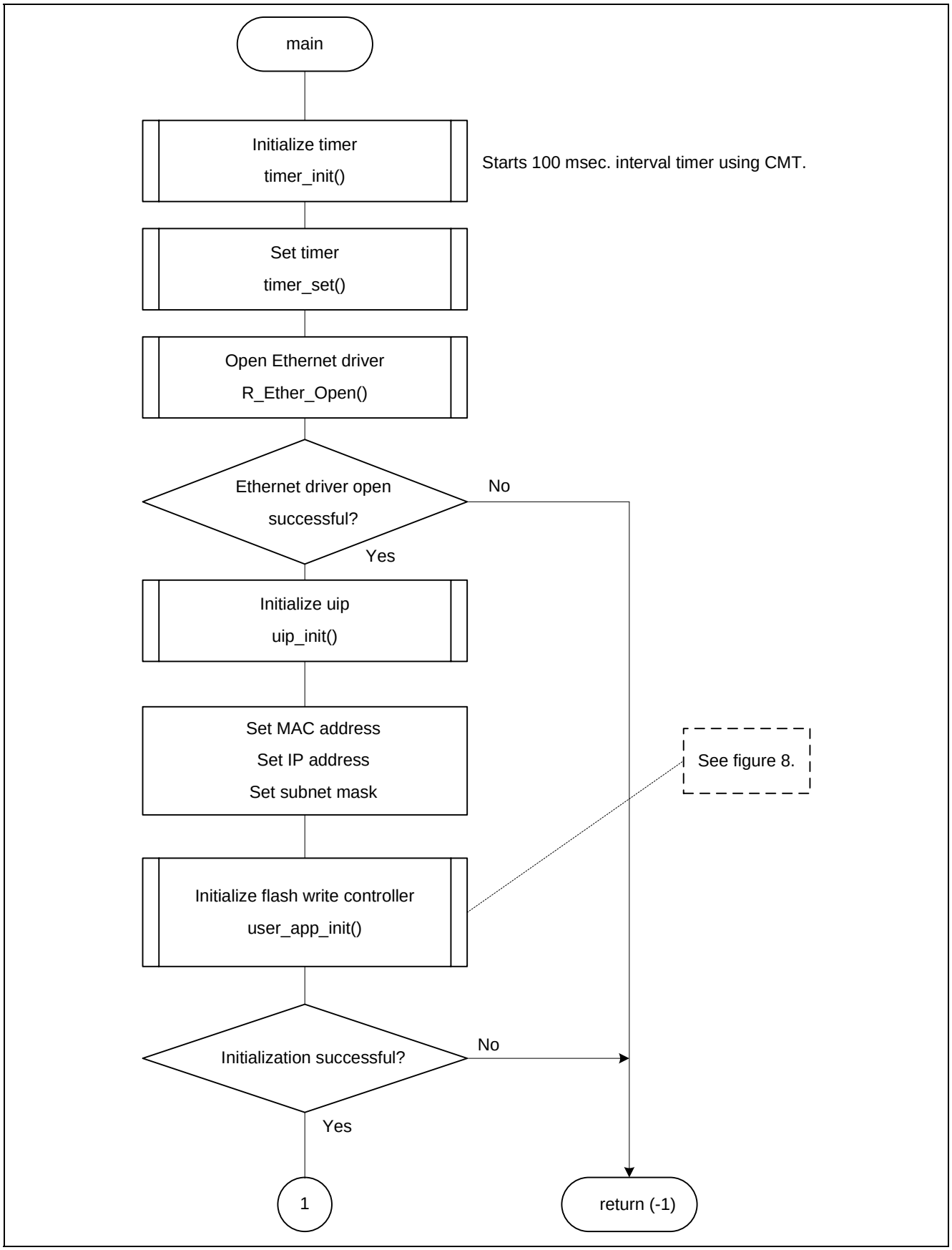


Figure 6 Main Process Flowchart (1/4)

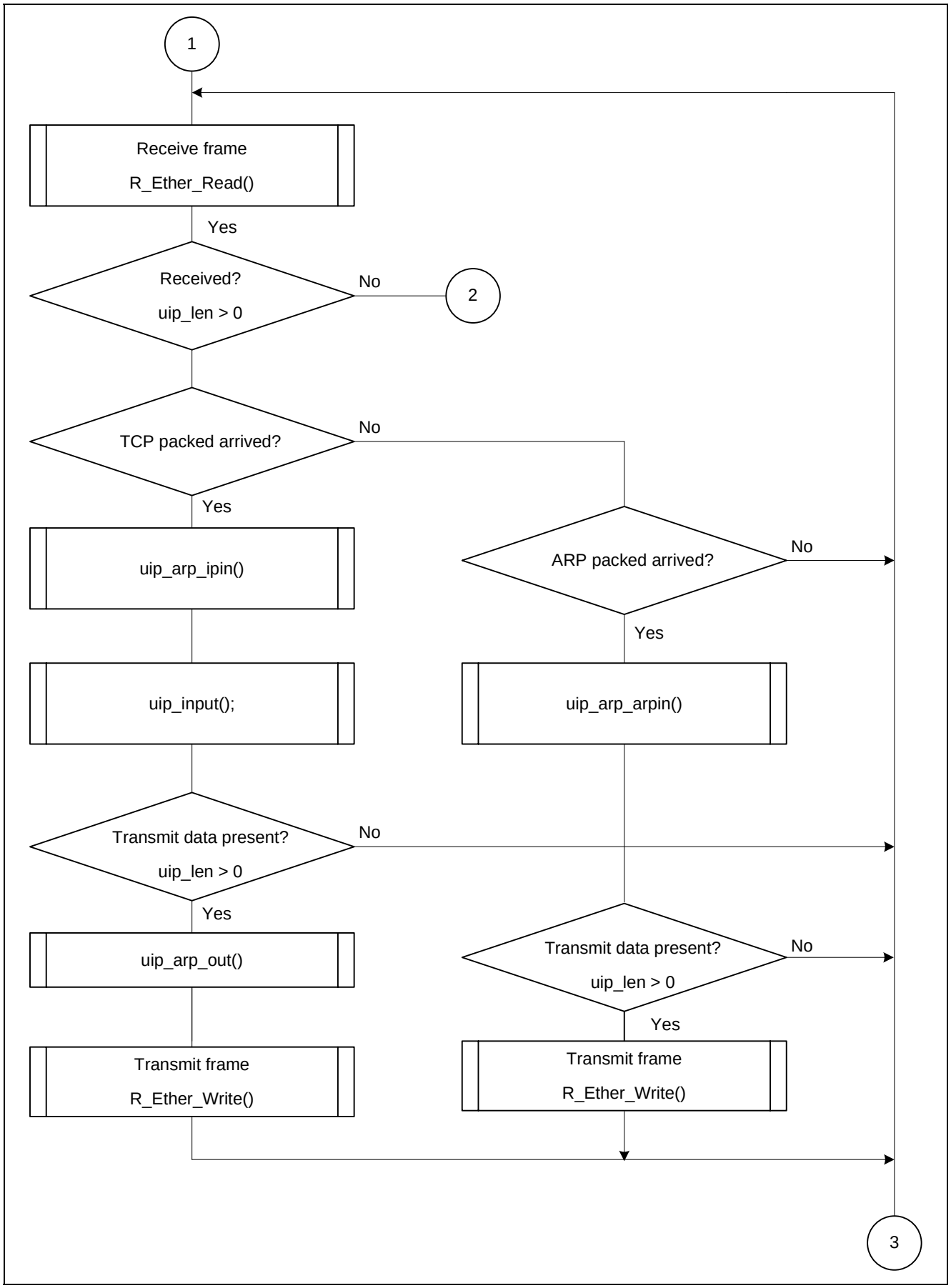


Figure 7 Main Process Flowchart (2/4)

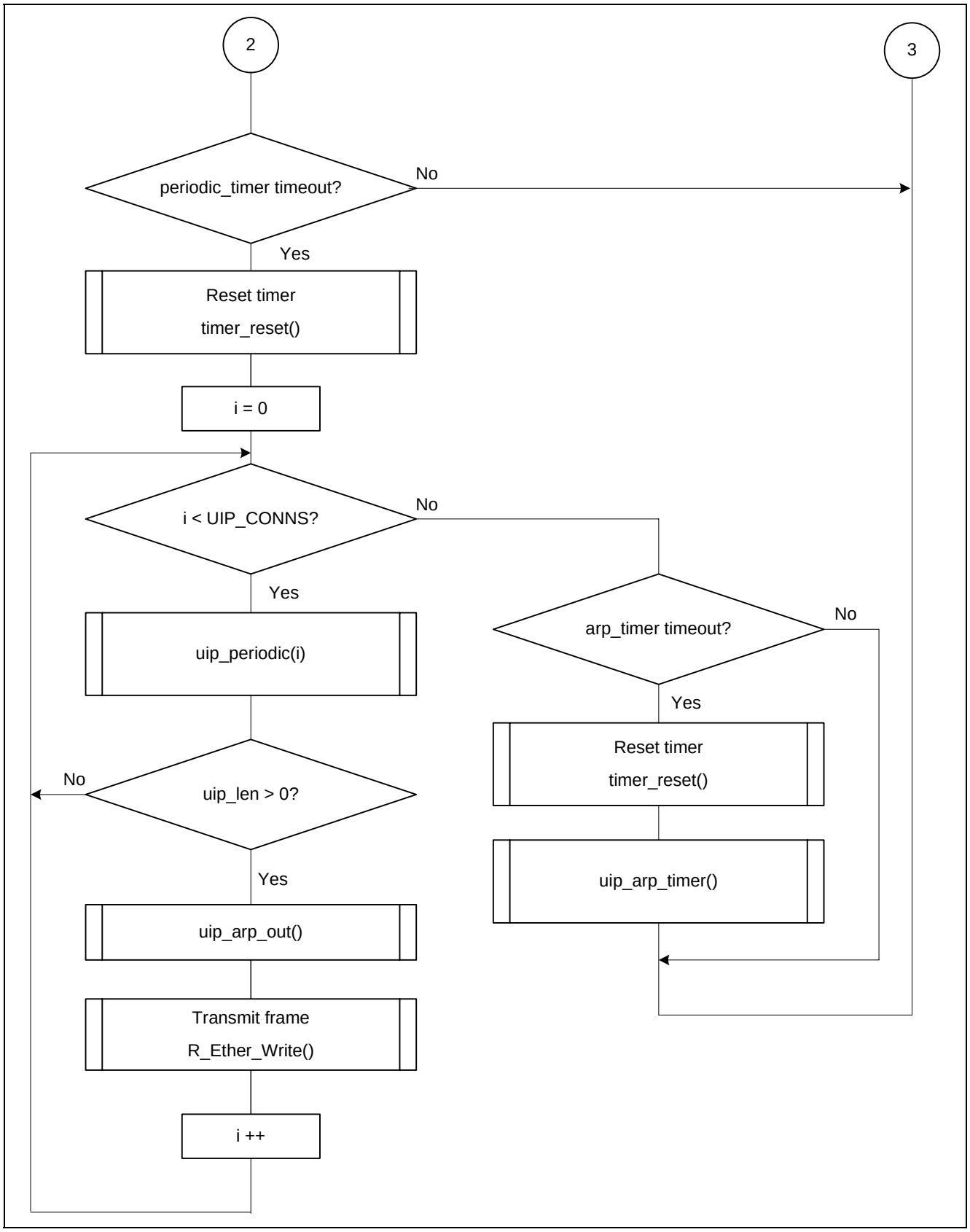


Figure 8 Main Process Flowchart (3/4)

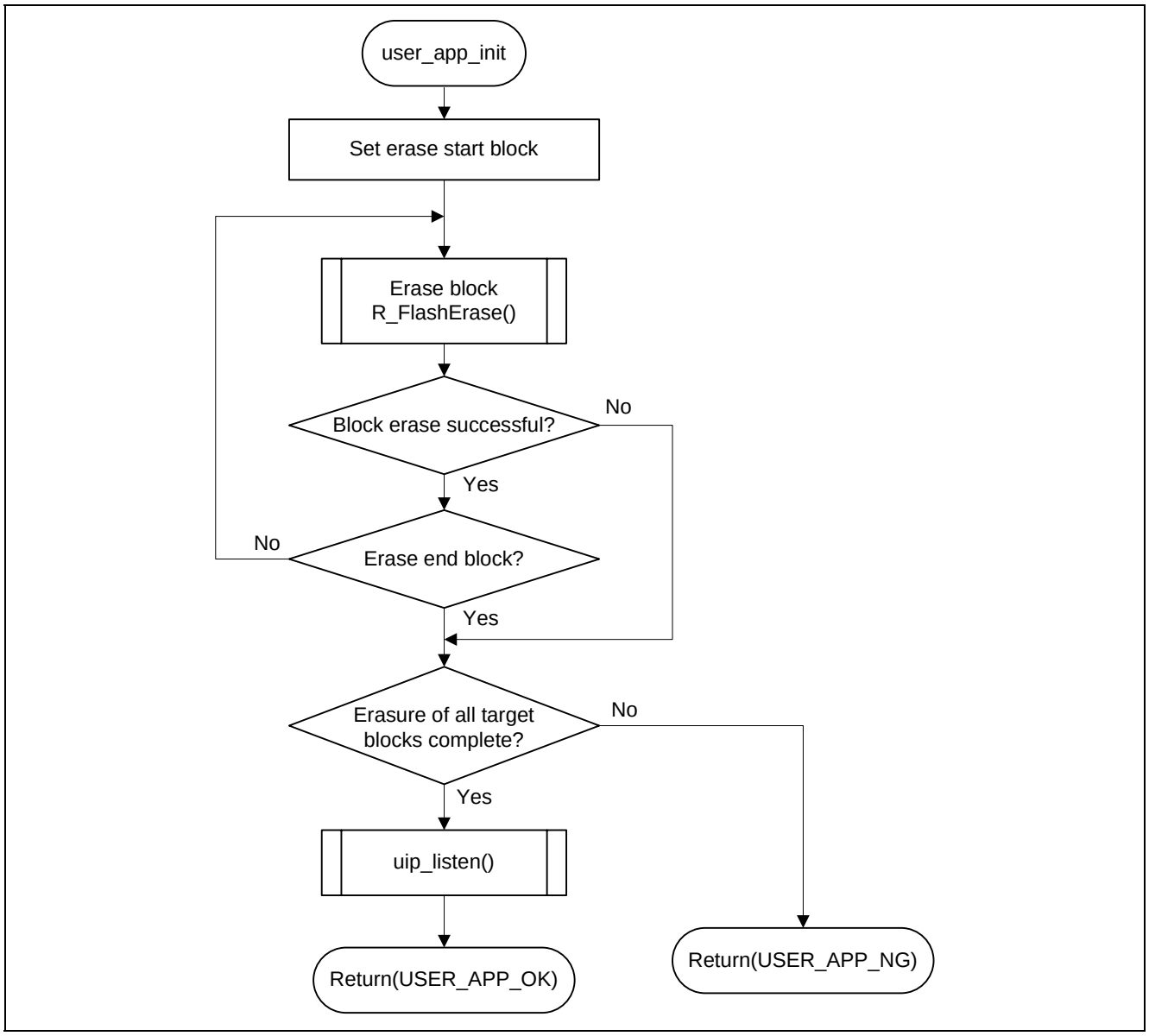


Figure 9 Main Process Flowchart (4/4)

2.2.2 Ethernet Driver

The Ethernet driver is a group of functions that uses the Ethernet controller (EtherC) and Ethernet controller direct memory access controller (E-DMAC) of the SH7216 to provide Ethernet frame transmit/receive functionality.

The SH7216 has a dedicated DMAC for the Ethernet controller (E-DMAC) to enable efficient transmitting and receiving of Ethernet frames. The E-DMAC can be controlled easily by using descriptors generated in memory.

Table 1 lists the functions opened by the Ethernet driver.

Table 1 List of Ethernet Driver Open Functions

No.	Function Name	Description
1	R_Ether_Open	Initializes EtherC, E-DMAC, and PHY.
2	R_Ether_Close	Closes EtherC and E-DMAC.
3	R_Ether_Read	Receives an Ethernet frame.
4	R_Ether_Write	Transmits an Ethernet frame.

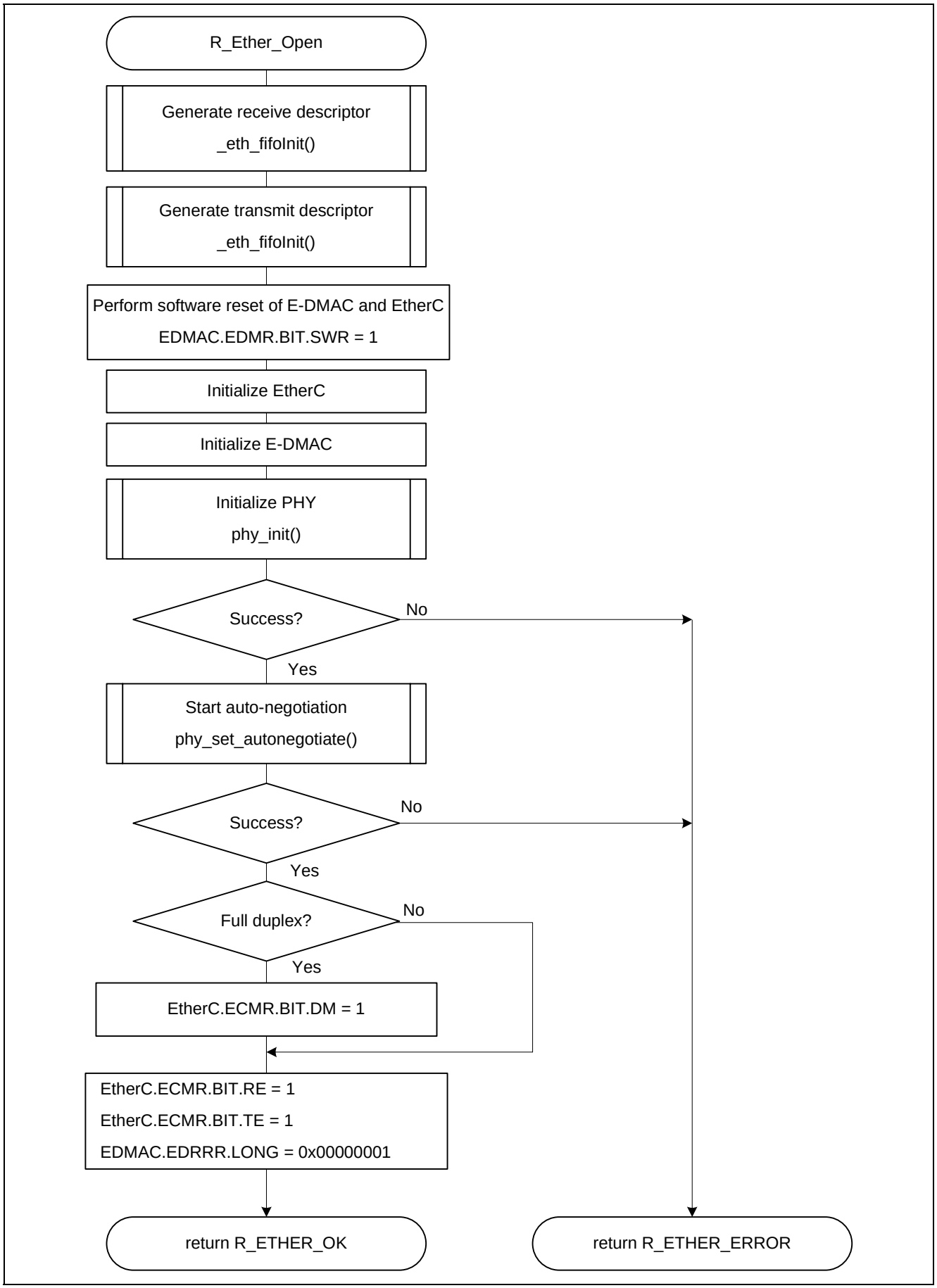


Figure 10 R_Ether_Open Flowchart

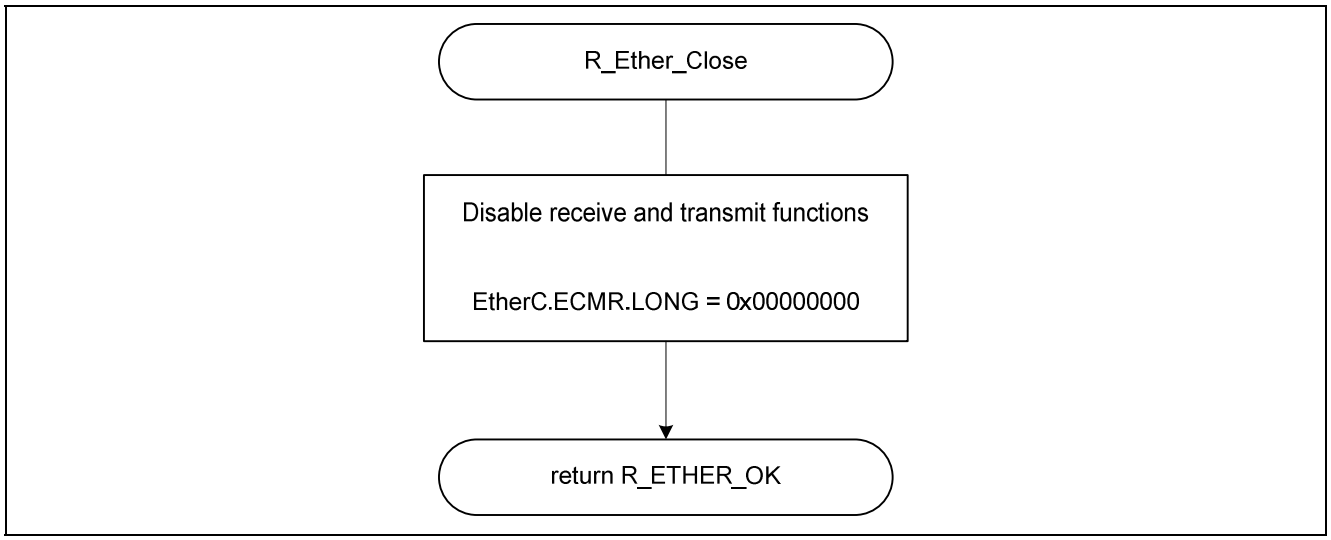


Figure 11 R_Ether_Close Flowchart

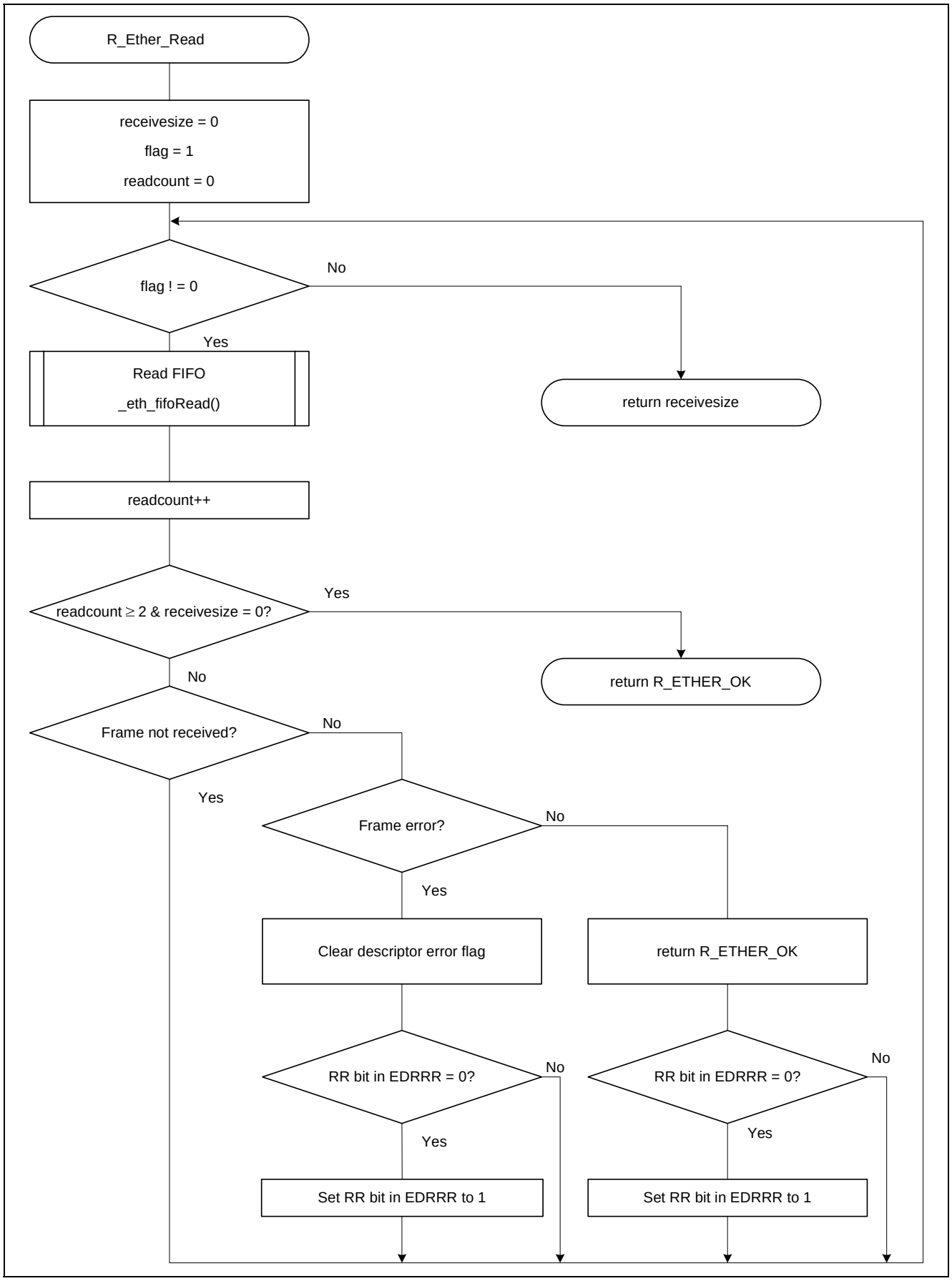


Figure 12 R_Ether_Read Flowchart

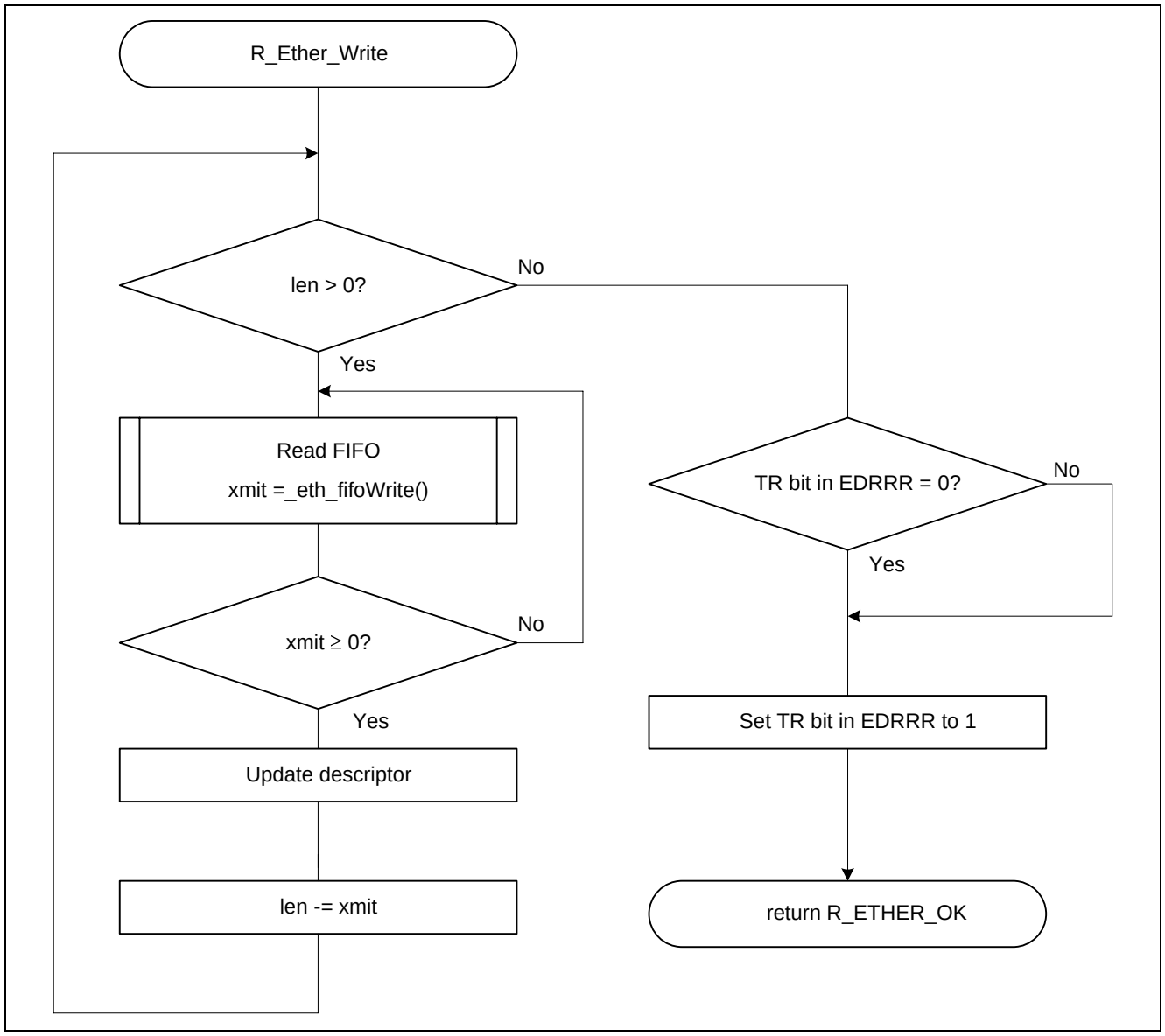


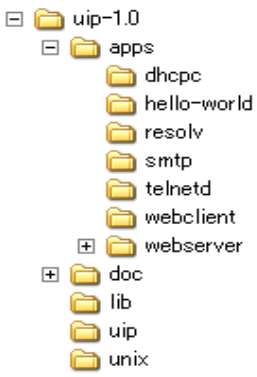
Figure 13 R_Ether_Write Flowchart

2.2.3 TCP/IP Protocol Stack

The sample application uses uip-1.0, which is open-source software, as the TCP/IP protocol stack.

Developed by Adam Dunkels, uip is a TCP/IP protocol stack designed for 8- and 16-bit MCUs. The code size of uip is only a few kilobytes and it only uses a few hundred kilobytes of memory, making it suitable for systems with limited resources.

Figure 14 shows the folder structure of uip-1.0.



apps:	This folder contains TCP/IP application samples that run on uip.
doc:	This folder contains the reference manual and sample code.
lib:	This folder contains libraries for securing memory blocks for TCP/IP applications.
uip:	This folder contains the TCP/IP protocol stack itself.
unix:	This folder contains sample code ported to Unix.

Figure 14 Folder Structure of uip-1.0

The sample application only uses the uip folder.

2.2.4 Programming and Erasing Flash Memory by Using Standard API

In the sample application, a standard API supplied by Renesas Electronics is used to program and erase the on-chip flash. Using the standard API makes it possible to program and erase the on-chip flash simply by calling the necessary functions from the user program.

Table 2 lists the standard API functions used by the sample application.

Table 2 List of Open Functions of Flash Memory Driver

No.	Function Name	Description
1	R_FlashWRITE	Writes data in 256-byte units.
2	R_FlashERASE	Erases data in block units.

Note: For details of the standard API, see the related application note (*SH Family: Simple Flash API for SH-2 and SH-2A*).

The user MAT is divided into eight 8 KB blocks, nine 64 KB blocks, and three 128 KB blocks. Programming is done in 256-byte units. For details of the block configuration, see figure 27.3 in the SH7216 Group Hardware Manual.

2.2.5 Flash Write Controller

The flash write controller is a program that corresponds to the application layer of the TCP/IP model. The program operates as a server application, and when launched it maintains a connection on port number 1000 (by default). When the host PC sends a connect request, the program waits to receive data transferred using TCP. Once 256 bytes of data (the number of bytes in a ROM write unit) has been received, the program writes it sequentially to the user MAT, starting at EB05 (H'0000A000).

The program uses the Protosockets library, a socket interface opened by uip, to implement data transfer using TCP.

The Protosockets library provides a socket interface similar to the BSD socket API. Some elements of the socket interface are blocked after calling a function such as recv(), and some cannot be implemented in a single-thread environment with no OS. The Protosockets library uses a module called Protothreads to overcome these limitations.

Protothreads is a module that was developed by Adam Dunkels, the developer of uip. It makes it possible to perform programming in a single-thread environment with no OS as if it were a multithreaded environment.

Figure 15 shows an sample code listing of a thread using Protothreads.

```
01: #include "pt.h"
02:
03: struct pt pt;
04: struct timer timer;
05:
06: PT_THREAD(example(struct pt *pt))
07: {
08:   PT_BEGIN(pt);
09:
10:   while(1) {
11:     if(initiate_io()) {
12:       timer_start(&timer);
13:       PT_WAIT_UNTIL(pt,
14:          io_completed() ||
15:          timer_expired(&timer));
16:       read_data();
17:     }
18:   }
19:   PT_END(pt);
20: }
```

Figure 15 Protothreads Sample Code

It is necessary to declare PT_BEGIN() at the beginning of the thread entry function and PT_END() at the end.

In the code sample above, a macro PT_WAIT_UNTIL, which causes processing to wait, is called in line 13. However, in actual operation control returns to PT_THREAD(). Then, the next time PT_THREAD() is called, execution proceeds from PT_WAIT_UNTIL. For this reason, it is necessary to call PT_THREAD() at regular intervals from the main loop.

The main loop of the flash write controller is declared as a single independent thread in Protothreads.

Figure 16 shows a flowchart of the main loop of the flash write controller.

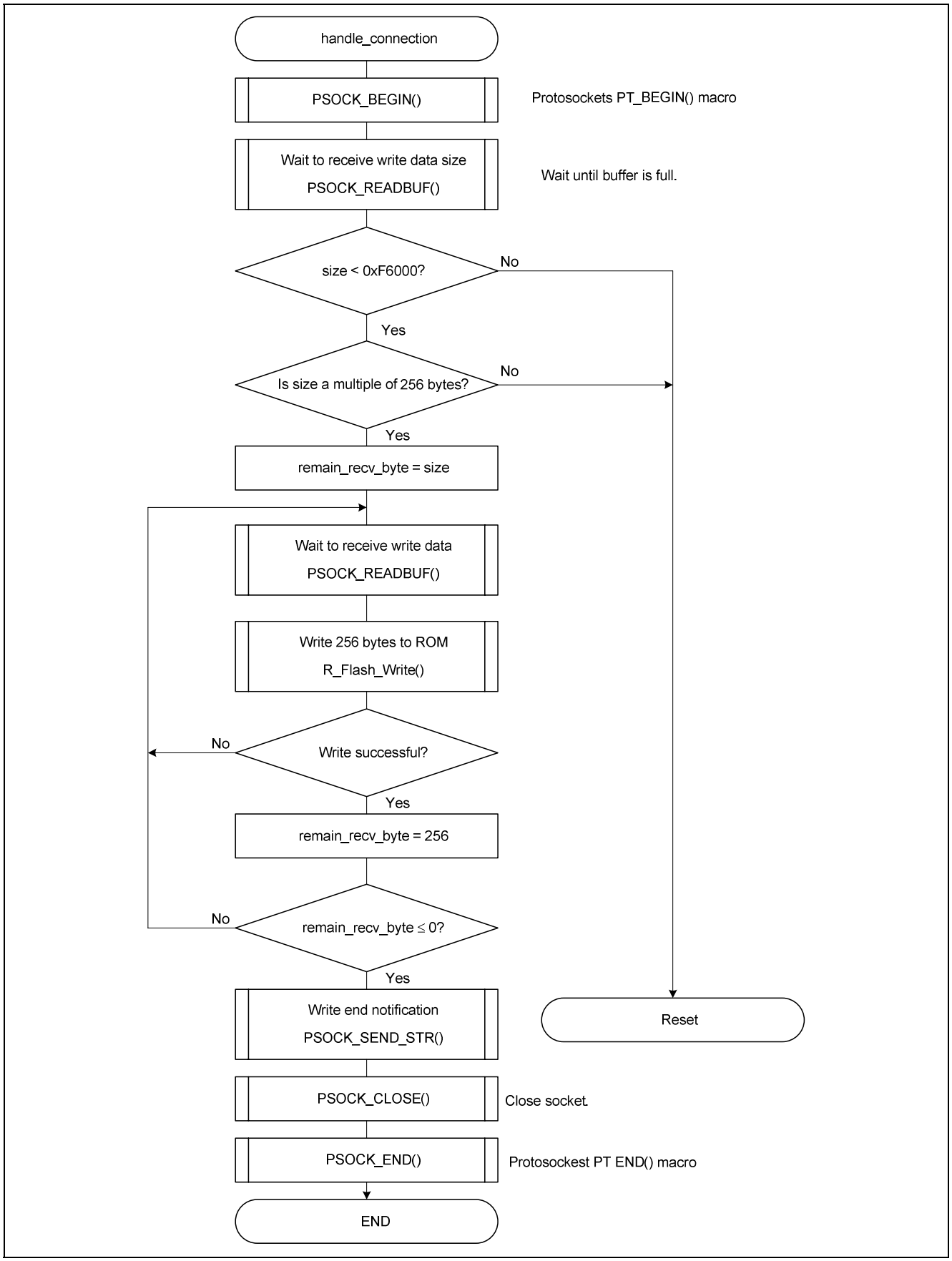


Figure 16 Flowchart of Main Loop of Flash Write Controller

2.2.6 Section Settings

Table 3 lists the sections of the sample program for the SH7216.

Table 3 Sections of Sample Program for SH7216

No.	Address	Section Name	Description
1	0x00000000	DVECTTBL	Vector table
2		DINTTBL	
3		PResetPRG	Reset handler
4		PIntPRG	Exception handler
5	0x00000A00	P	Program area
6		PFRAM	Flash memory driver
7		C	Constant area
8		C$BSEC	
9		C$DSEC	
10		D	Initialization data area
11	0xFFF80000	RDVECTTBL	Vector table (allocated in RAM)
12		RDINTTBL	
13		RPResetPRG	Reset handler (allocated in RAM)
14		RPIntPRG	Exception handler (allocated in RAM)
15	0xFFF80A00	RP	Program area (allocated in RAM)
16		RPFRAM	Flash standard API (allocated in RAM)
17		RC	Constant area (allocated in RAM)
18		RC$BSEC	
19		RC$DSEC	
20		R	Initialization data area (allocated in RAM)
21	0xFFF88F00	B	Uninitialized data area
22	0xFFF8DA00	B_RX_DESC	Descriptor area for Ethernet
23		B_TX_DESC	Must be allocated at 16-byte boundary.
24		B_RX_BUFF	Buffer area for Ethernet
25		B_TX_BUFF	
26	0xFFF8F800	S	Stack area

The program must be run from RAM because it erases and programs the user MAT in the ROM. Therefore, numbers 1 to 10 in the user MAT in the ROM are transferred to numbers 11 to 20 in RAM immediately after the power-on reset startup, and the program runs from RAM from that point forward.

2.3 Sample Program for Host PC

The sample program for the host PC is a program that corresponds to the application layer of the TCP/IP model and operates as a client application

The program is a console application. When it is launched, the user specifies the server IP address, server port number, and write data file name. The program then attempts to connect to the specified IP address and port number. When it succeeds, it transmits by TCP the write data file in the user MAT in the ROM to the sample program for the SH7216. The maximum size of the write data file is 1,007,616 bytes.

The program uses the Winsock library to implement data transfer using TCP.

Winsock is a technical specification that defines the method used by Windows network software to access network services. It defines a standard interface between TCP/IP client applications running on Windows and the TCP/IP protocol stack. The interface is based on the socket API model used for program communication in BSD UNIX.

The operating environment and development environment of the program are shown in table 4.

Table 4 Operating Environment and Development Environment of Program

Operating environment	CPU	x86
	OS	Microsoft Windows XP
Development environment	Microsoft Visual Studio 2005	

Figures 17, 18, and 19 show a flowchart of the sample program that runs on the host PC.

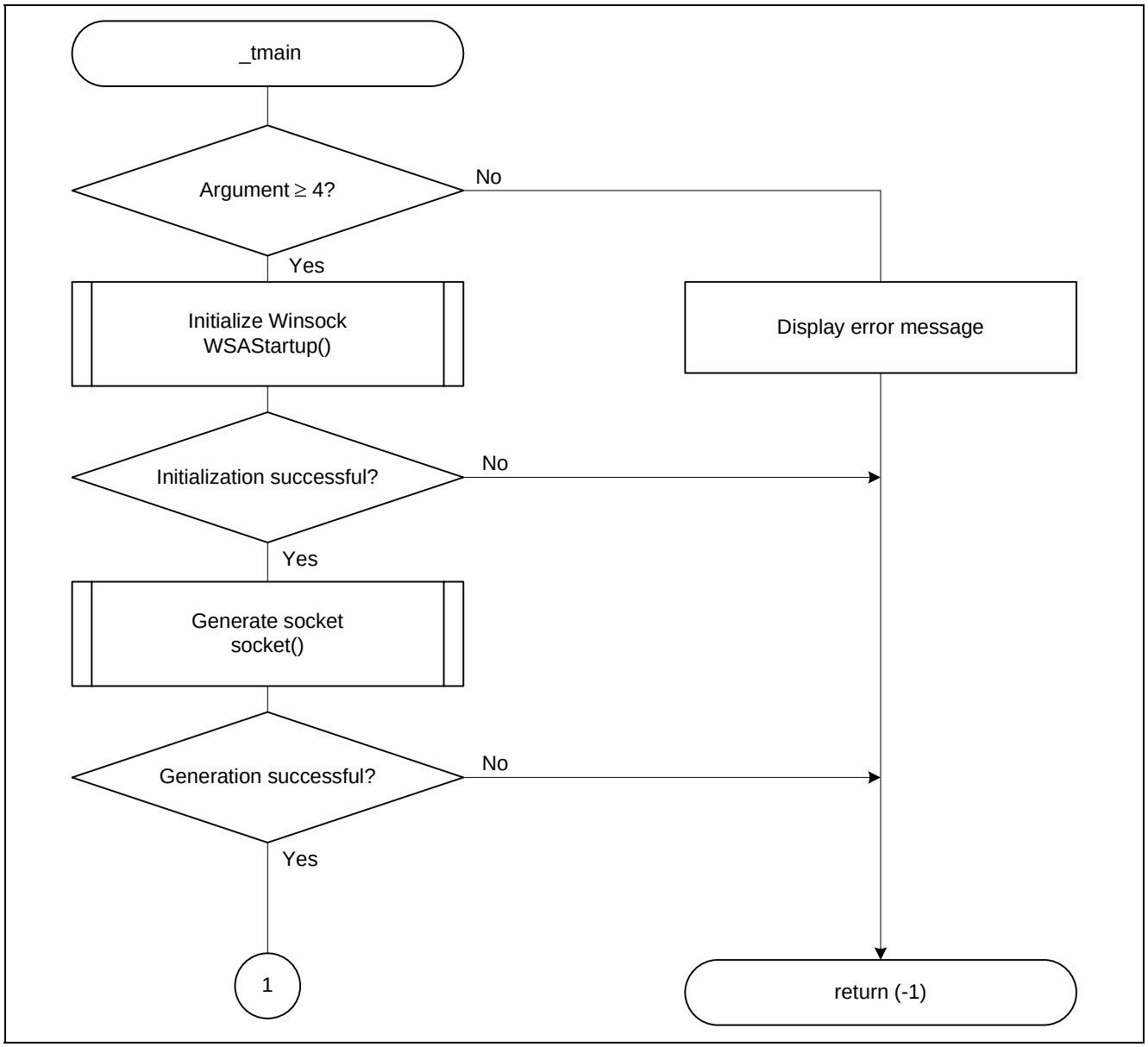


Figure 17 Flowchart of Host PC Sample Program (1/3)

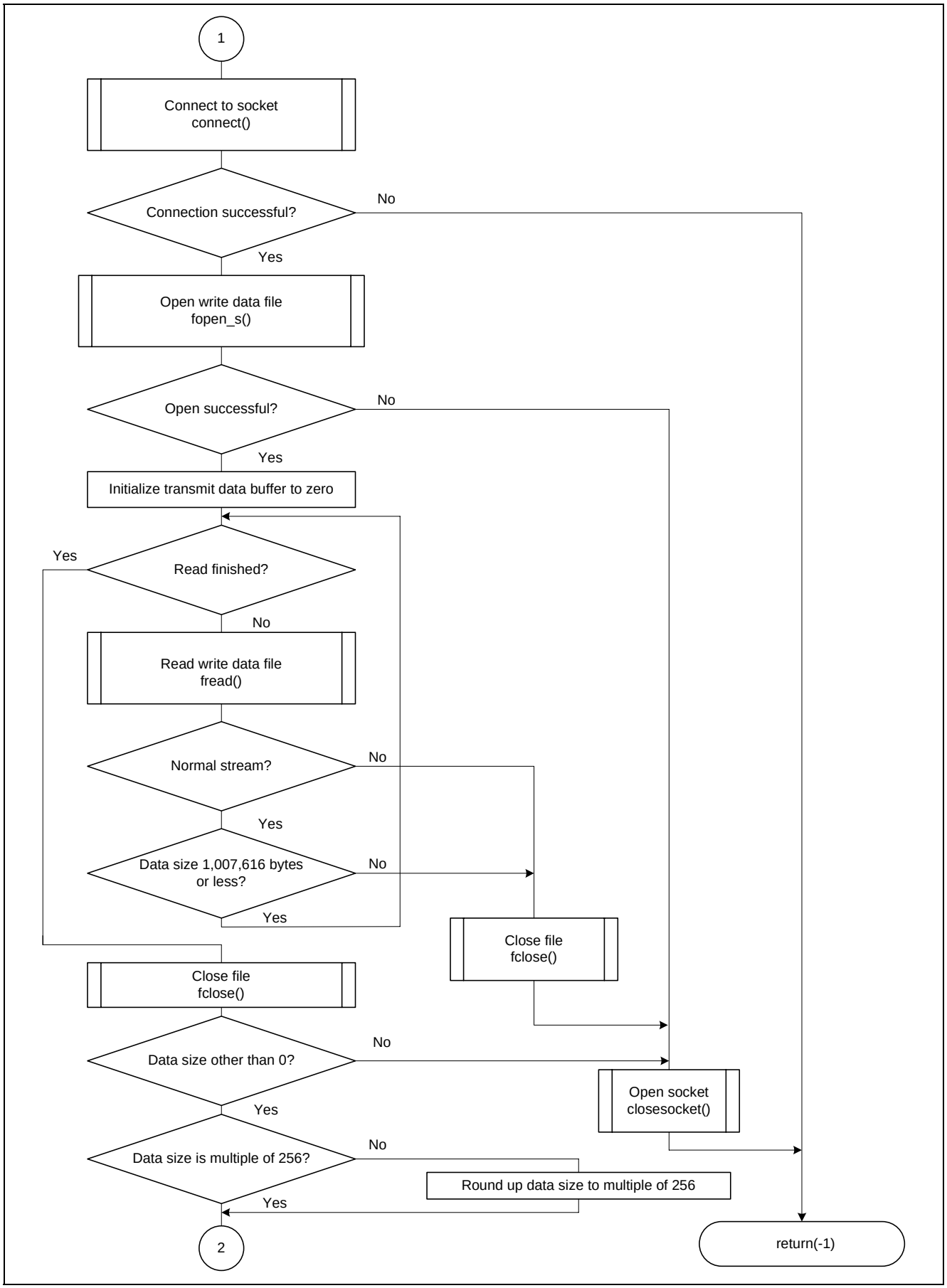


Figure 18 Flowchart of Host PC Sample Program (2/3)

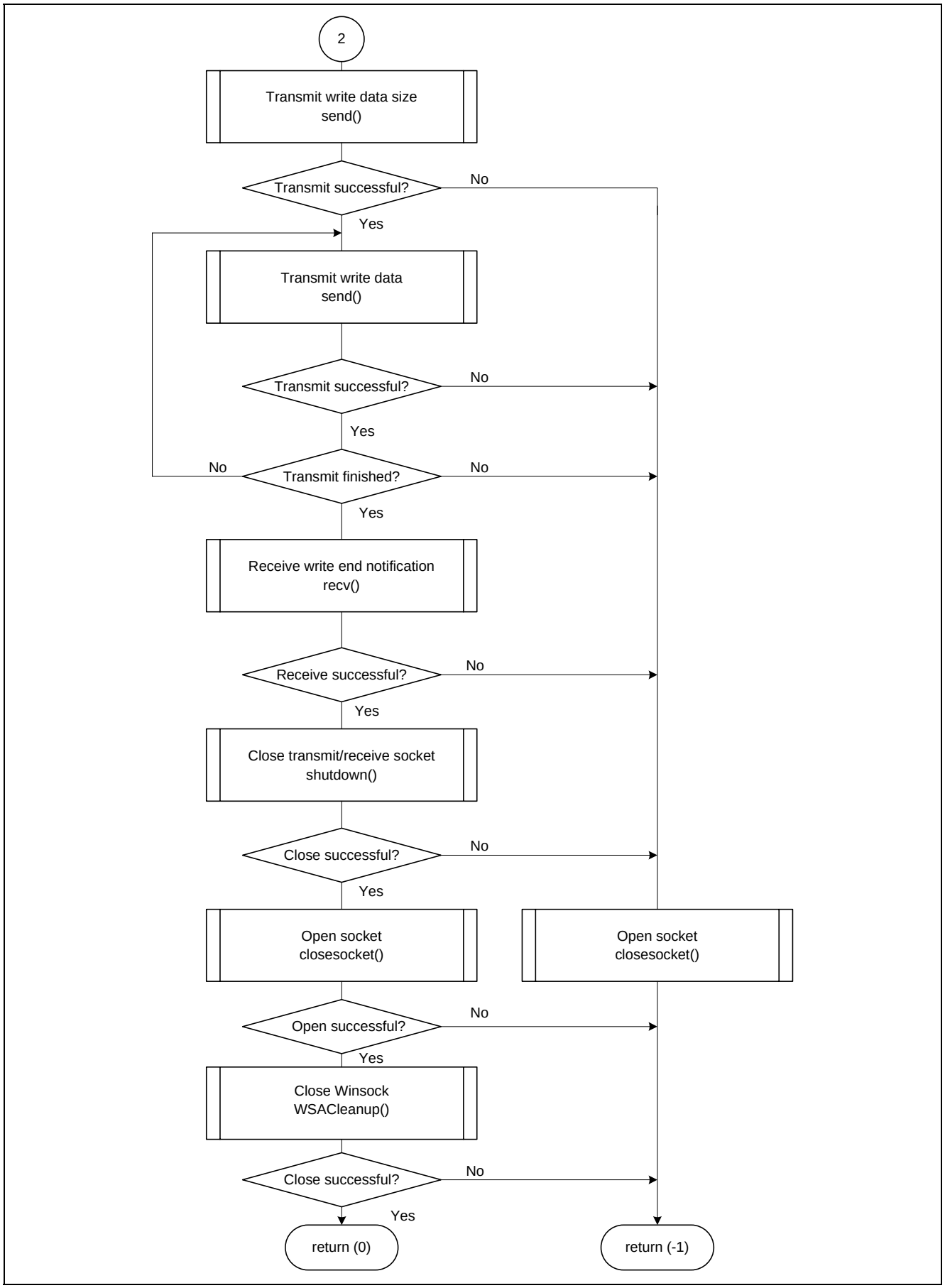


Figure 19 Flowchart of Host PC Sample Program (3/3)

3. Using the Sample Program

This section describes how to use the sample program for the SH7216 and the sample program for the host PC.

Figure 20 shows a device configuration diagram.

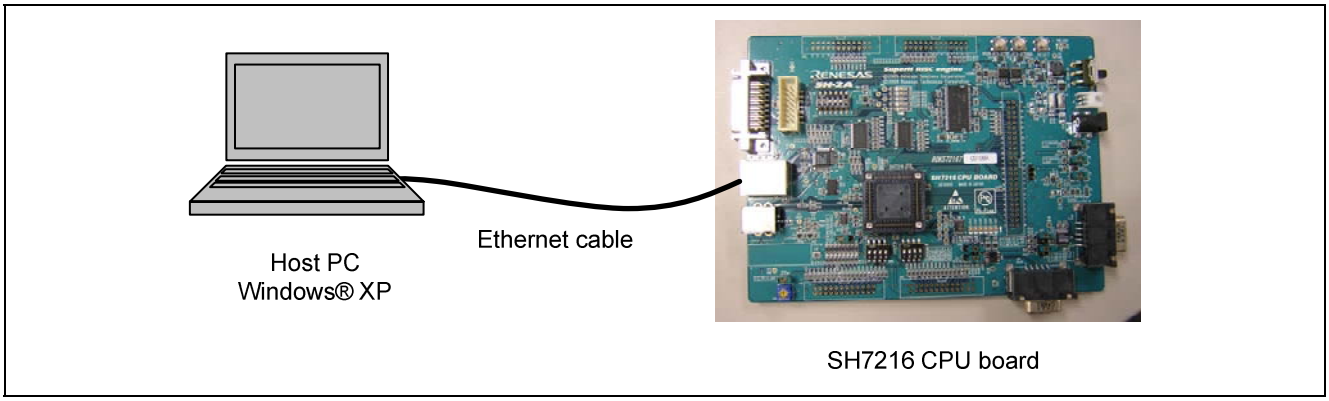


Figure 20 Device Configuration Diagram

Before performing the steps below, confirm that the switches on the SH7216 CPU board are in their default setting positions. For information on the default settings, see 3.2.6, Switch and LED Functions, in the *SH7216 CPU Board R0K572167C001BR User's Manual, Rev. 0.03*.

3.1 Usage

Perform the steps below to check operation using the PC.

Note: Make sure to write the sample code to the on-chip flash before performing the steps below.

1. Set the IP address and subnet mask of the host PC to the following values.
   IP address: 192.168.1.77
   Subnet mask: 255.255.255.0
2. Connect the PC to the SH7216 CPU board using an Ethernet cable as shown in figure 20, above.
3. Open a command prompt window and navigate to the location of the sample program for the host PC (SH7216-test_client.exe).
   After navigating to the program location, launch the sample program for the host PC (SH7216-test_client.exe), specifying as arguments "192.168.1.76" for server IP address, "1000" for server port number, and "<user defined file name>" as the write data file name.
   The operation are completed successfully. Press the Enter key to quit the program.

4. Notes on Use of Flash Memory (ROM)

There are important points to be borne in mind when using the flash memory (ROM).

For details, see 27.10, Usage Notes, in the *SH7216 Group Hardware Manual* for details.

5. Reference Documents

- Hardware Manual
  SH7216 Group Hardware Manual, Rev. 1.01
  (The latest version can be downloaded from the Renesas Electronics Web site.)
- SH716 CPU Board User's Manual
  SH7216 CPU Board R0K572167C001BR User's Manual, Rev. 0.03
- uip 1.0 Reference Manual
  The uIP Embedded TCP/IP Stack: The uIP 1.0 Reference Manual (June 2006)

Website and Support

Renesas Electronics Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/inquiry

All trademarks and registered trademarks are the property of their respective owners.

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Apr.12.10	—	First edition issued
2.00	Dec.14.10	—	Standard API used to program and erase on-chip flash
2.10	Mar.17.11	—	Added read after FRQCR settings
2.11	Apr.06.11	—	Host tool updated

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins
   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.
   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.
2. Processing at Power-on
   The state of the product is undefined at the moment when power is supplied.
   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.
3. Prohibition of Access to Reserved Addresses
   Access to reserved addresses is prohibited.
   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.
4. Clock Signals
   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.
   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.
5. Differences between Products
   Before changing from one product to another, i.e. to one with a different type number, confirm that the change will not lead to problems.
   — The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
   - "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.
   - "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
   - "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

RENESAS

SALES OFFICES

Renesas Electronics Corporation

http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
7F, No. 363 Fu Shing North Road Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

© 2011 Renesas Electronics Corporation. All rights reserved.

Colophon 1.0